(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 December 2002 (19.12.2002)

PCT

(10) International Publication Number
WO 02/101493 A2

(51) International Patent Classification[7]: G06F

(21) International Application Number: PCT/US02/17751

(22) International Filing Date: 7 June 2002 (07.06.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/296,113	7 June 2001 (07.06.2001)	US
60/296,117	7 June 2001 (07.06.2001)	US
60/296,118	7 June 2001 (07.06.2001)	US
60/331,621	20 November 2001 (20.11.2001)	US
60/331,623	20 November 2001 (20.11.2001)	US
60/331,624	20 November 2001 (20.11.2001)	US
60/331,625	20 November 2001 (20.11.2001)	US

(71) Applicant: **CONTENTGUARD HOLDINGS, INC.** [US/US]; 103 Foulk Road, Suite 200-M, Wilmington, DE 19803 (US).

(72) Inventors: **WANG, Xin**; 3005 Shrine Place, #8, Los Angeles, CA 90007 (US). **DEMARTINI, Thomas**; 6410 Green Valley Circle #130, Culver City, CA 90230 (US). **CHEN, Eddie**; 6796 Vallon Drive, Rancho Palos Verdes, CA 90275 (US). **GILLIAM, Charles, P.**; 27 Beach Drive, Darien, CT 06820 (US). **HAM, Manuel**; 10259 Pangborn Avenue, Downey, CA 90241 (US). **LAO, Guillermo**; 5531 Lorna Street, Torrance, CA 90503 (US). **RALEY, Michael**; 12834 Verdura Avenue, Downey, CA 90242 (US). **TA, Thahn**; 18694 Stratton Lane, Huntington Beach, CA 92648 (US). **TADAYON, Bijan**; 20920 Scottsbury Drive, Germantown, MD 20876 (US).

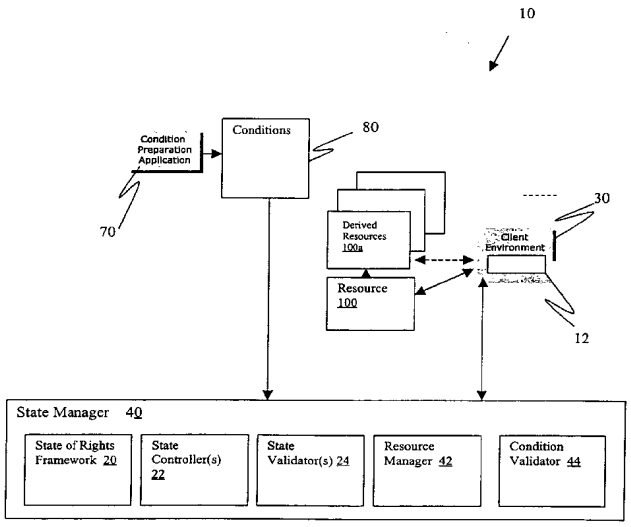
(74) Agent: **SONG, Daniel, S.**; Nixon Peabody LLP, Suite 800, 8180 Greensboro Drive, McLean, VA 22102 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR TRACKING STATUS OF RESOURCE IN A SYSTEM FOR MANAGING USE OF THE RESOURCES



(57) Abstract: A method and system for managing the state of a protected resource in a system for granting access to a protected resource in accordance with the usage rights. The usage rights include at least one state variable indicating a status of an associated protected resource. A message related to the state variable is transmitted from a resource control device to an interface framework. The resource control device is coupled to the resource ton control use of the resource by enforcing the usage right. A state controller operative to track the value of the state variable is loaded into the framework and instructed to manipulate the value of the state variable in accordance with said message. For example, the message can be a query of the current value of the state variable.

WO 02/101493 A2

Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND APPARATUS FOR TRACKING STATUS OF RESOURCE IN A SYSTEM FOR MANAGING USE OF THE RESOURCES

COPYRIGHT NOTICE

BACKGROUND

Field of the Invention:

[0002] The invention relates to the management of resources in a computer system. More specifically, the invention relates to a method and apparatus for tracking the state of protected resources, such as digital content, and usage rights associated with the protected resources.

Description of the Related Art

BACKGROUND

[0003] One of the most important issues impeding the widespread distribution of digital works (i.e. documents or other content in forms readable by computers), via electronic means, and the Internet in particular, is the current lack of ability to enforce the intellectual property rights of content owners during the distribution and use of digital works. Efforts to resolve this problem have been termed "Intellectual Property Rights Management" ("IPRM"), "Digital Property Rights Management" ("DPRM"), "Intellectual Property Management" ("IPM"), "Rights Management" ("RM"), and "Electronic Copyright Management" ("ECM"), collectively referred to as "Digital Rights

Management (DRM)" herein. There are a number of issues to be considered in effecting a DRM System. For example, authentication, authorization, accounting, payment and financial clearing, rights specification, rights verification, rights enforcement, and document protection issues should be addressed. U.S. patents 5,530,235, 5,634,012, 5,715,403, 5,638,443, and 5,629,940, the disclosures of which are incorporate herein by reference, disclose DRM systems addressing these issues.

[0004] For example, U.S. patent 5,634,012, discloses a system for controlling the distribution of digital documents. Each rendering device has a repository associated therewith. A predetermined set of usage transaction steps define a protocol used by the repositories for enforcing usage rights associated with a document. Usage rights persist with the document content. The usage rights specify various manners of use of the content such as, viewing only, use once, distribution, and the like. Pre-conditions, such as payment of a fee, proof of identity or other conditions can be required prior to permitting access to the content in accordance with the usage rights. Once the pre-conditions are satisfied access to the content is granted. The concept of conditional access is well known in access control applications also. For example, it is known to grant access to network resources upon entry of login name and password.

[0005] The concept of conditional access is a foundation for both access control and DRM systems. A typical pre-condition, i.e. a condition for granting access, defines a list of authorized users along with a set of access rights and conditions to a given resource. Pre-conditions associated with a given resource can be defined as resources associated with certain users. This is known as "role-based" access control. Pre-conditions can also be defined by rules in a process known as "rule-based" access control. Both types of pre-conditions are expressed as an access control list, which is a set of resources or rules defined in some language or data structure.

[0006] Conditional access is typically implemented by most systems as an authorization process in which a principal (e.g., a person, a system or a process) is allowed access to a protected resource after certain conditions are met and/or verified.

SUMMARY OF THE INVENTION

[0007] A first aspect of the invention is a system for managing the state of a protected resource in a system for granting access to a protected resource in accordance with usage rights. The usage rights include state variables indicating a status of an associated protected resource. The system comprises a protected resource associated with a usage right defined at least in part by a state variable, a resource control device coupled to said resource to control use of said resource by enforcing the usage right, a state controller operative to track the value of a state variable, and an interface framework operative to receive a message related to said state variable from said resource management device, load said state controller, and instruct said state controller to manipulate the value of the state variable in accordance with said message.

[0008] A second aspect of the invention is a method for managing the state of a protected resource in a system for granting access to a protected resource in accordance with usage rights including a state variable indicating a status of an associated protected resource. The method comprises transmitting a message related to the state variable from a resource control device to an interface framework, said resource control device being coupled to said resource to control use of said resource by enforcing the usage right, loading into said framework a state controller operative to track the value of the state variable, and instructing said state controller to manipulate the value of the state variable in accordance with said message.

BRIEF DESCRIPTION OF THE DRAWING

[0009] The invention is described through a preferred embodiment and the attached drawing in which:

[0010] Fig. 1 is a block diagram of a computer architecture of the preferred embodiment;

[0011] Fig. 2 is a schematic illustration of the states of a conventional access control model;

[0012] Fig. 3 is a schematic illustration of the states of the preferred embodiment;

[0013] Fig. 4 is a schematic illustration of a condition/right of the preferred embodiment;

[0014] Fig. 5 is a schematic illustration of a state of rights of the preferred embodiment; and

[0015] Fig. 6 is a schematic illustration of the state manager of the preferred embodiment with detail of the state of rights framework; and

[0016] Fig. 7 is a schematic illustration of the state manager with detail of the resource manager.

DETAILED DESCRIPTION

[0017] Different types of resources require different types of conditions and different mechanisms to protect them from unauthorized use. In the preferred embodiment conditions and rights are part of the entire lifecycle of protected resources. This means that conditions of rights are not just evaluated prior to allowing access, but are also evaluated during the actual consumption of the resource. Additionally, conditions are associated with both the protected resource and the state of the protected resource. Associating conditions with various states of a protected resource provides content owners or service providers a flexible way to protect different types of resources. Resources

can be digital content, hardware, software programs, memory space, goods, services (including web services), time, fees, usage rights or a license.

[0018] Usage rights, specify manners of use. For example, a manner of use can include the ability to use an item in a specified way for a specified period of time. Further, usage rights can specify transfer rights, such as distribution rights and can permit granting of usage rights to others or the derivation of usage rights. In the preferred embodiment conditions and rights are associated to define manners of use of resources. Accordingly, conditions and rights are referred to as a single entity in the preferred embodiment. However, conditions and rights can exist separately.

[0019] The preferred embodiment verifies and validates conditions of rights prior, during and after the usage or consumption of protected resources. Rights and associated conditions can be represented as a state so that the current state and history of each right can be logged and later used. "State variables" track potentially dynamic conditions. State variables are variables having values that represent status of a resource or other dynamic conditions. State variables can be tracked, and the value of state variables can be used in a condition. For example, a usage right, as a resource, can be the right to view content and a condition can be that no other users are logged into the network when the usage right is exercised. In this example, when the value of the appropriate state indicates that other users are logged in, the condition is no longer satisfied and the content can not be viewed, or viewing is terminated.

[0020] Fig. 1 illustrates computer architecture 10 of the preferred embodiment. Conditions 80 are described in detail below and can be prepared with preparation application 70 associated with the distributor of an item, a content service provider, an enterprise manager or any other party wishing to control access to resources, such as digital content. A grammar such as XrML™ can be used to specify conditions 80. However, conditions 80 can be specified in any manner. A user operates within client environment

30, including a computer or other device associated with the user. Software application 12, such as a rendering engine or other application, can be installed in client environment 30. State manager 40 controls access to protected resource(s) 100 and derived resource(s) 100a in the manner set forth below.

[0021] State manager 40, a computer device in the preferred embodiment, addresses security aspects of access to resource 100 and derived resource 100a. In particular, state manager 40 may authenticate messages by verifying and validating signatures, such as a cryptographic signature, or other identifying characteristics of the messages in a known manner. State manager 40 includes, resource manager 42 and condition validator 44. Resource manager 42 is responsible for resource registration, resource transformation, and resource termination. "Transformation" refers to deriving derived resource 100a from resource 100. As illustrated in Fig. 7, transformation is accomplished by resource transformation module 48 of resource manager 42. For example, in the case where the resource is an encrypted file which represents an image or the like, the derived resources 100a could include the clear image itself and the address of the memory that holds the image. During resource registration the memory address that holds the image is recorded by resource repository 46 of resource manager 42, so that any access to that memory, i.e. derived resource 100a, can be tracked. In addition, tracking marks (such as a watermark) can be inserted into the image, so that it can be tracked at any time.

[0022] Condition validator 44 monitors the set conditions and manages the "state of the rights", i.e., a collection of current values of state variables for a given right, as described below. Condition validator 44 interacts with the resource manager 46, as described in greater detail below, to control derived resource 100a. When the current state of right 44b is no longer valid, i.e., does not correspond to authorized state of rights 44a condition validator 44 requests resource manager 42 to delete (or disable) all the derived resources 100a or to notify application 12 that the use of derived resources 100a is no

longer allowed as described in greater detail below. State Manager 40 also includes state of rights framework 20, state controller 22, and state validator 24, all of which are described in detail below.

[0023] Access to protected resource 100 is based on conditions 80. This type of condition is referred to as an access condition or "pre-condition." However, by associating conditions with both resource 100 and the state of resource 100, it becomes possible to protect resource 100 at various stages during its life cycle. Resource 100 can be protected before a user is granted access, when access is granted, during the actual use of resource 100 and after the use of resource 100. Conditions 80 that are associated with the entire lifecycle of the protected resource can be expressed by use of a grammar including data structures, sets of rules or a language such as XrML™. The preferred embodiment uses of XrML™ as the language to express conditions.

[0024] To protect resource 100, conditions 80 can be imposed on resource 100 itself, or any other resources – either tangible or intangible - including those resources that make up any executing environments, such as a application 12 of client environment 30, from which protected resource 100 is accessed and used.

[0025] Conditions 80 can be the identity of a user or a group of users who is/are granted access as principals and who use protected resource 100. Examples of conditions 80 are set forth below as expressed in the XrML™ language. Example A expresses a condition associated with a principal "Edgar" who is granted the right to "view" the protected digital content "XrML Book". Example B expresses a condition associated with a group of principals, i.e. all persons that fall under the category "ContentGuard employee" who are granted the right to print the protected digital work "XrML Book".

Example A:
<license>

```
        <inventory>
                <digitalWork licensePartID="XrMLBook"/>
                <keyHolder licensePartID="Edgar"/>
        </inventory>
        <grant>
                <keyHolder licensePartIDRef="Edgar"/>
                <view/>
                <digitalWork licensePartIDRef="XrMLBook"/>
        </grant>
</license>
```

Example B:

```
<license>
        <inventory>
                <digitalWork licensePartID="XrMLBook"/>
        </inventory>
        <grant>
                <forAll varName="ContentGuard Employee"/>
                        <principal varRef=" ContentGuard Employee"/>
                        <print/>
                        <digitalWork licensePartIDRef="XrMLBook"/>
        </grant>
</license>
```

[0026] Conditions 80 can be conditions on principals who must possess certain properties, such as a certain title, or rights, such as a security clearance. Example C expresses the condition that principals must possess a manager's badge.

Example C:

```
<license>
        <inventory>
                <digitalWork licensePartID="XrMLBook"/>
                <keyHolder licensePartID="Edgar"/>
        </inventory>
                <grant>
                        <forAll varName="anyone"/>
                        <principal varRef="anyone"/>
                        <possessProperty/>
                        <badge>
                        <title>Manager</title>
                        </badge>
                        <view/>
                        <digitalWork licensePartIDRef="XrMLBook"/>
```

```
        </grant>
</license>
```

[0027] Conditions 80 can be conditions on the time interval for access to the protected item. Example D below expresses conditions that key holder "Edgar", as a principal, can view the content "XrML book" not before 05/29/2002 and not after 05/29/2003.

Example D:

```
<license>
        <inventory>
                <digitalWork licensePartID="XrMLBook"/>
                <keyHolder licensePartID="Edgar"/>
        </inventory>
        <grant>
                        <keyHolder licensePartIDRef="Edgar"/>
                        <view/>
                        <digitalWork licensePartIDRef="XrMLBook"/>
                        <validityInterval>
                        <notBefore>2002-05-29T00:00:00</notBefore>
                        <notAfter>2003-05-29T00:00:00</notAfter>
                </validityInterval>
        </grant>
</license>
```

[0028] Conditions 80 can be related to the physical location of the principal or resource used to access content. Example E below expresses the condition that anyone currently in US can print the content "XrML Book"

Example E:

```
<license>
        <inventory>
                <digitalWork licensePartID="XrMLBook"/>
        </inventory>
        <grant>
                <forAll varName="anyone"/>
                <principal varRef="anyone"/>
                <print/>
                <digitalWork licensePartId="XrMLBook"/>
                <territory>
                        <country>US</country>
                </territory>
        </grant>
```

```
</license>
```

[0029] Conditions 80 can specify a fee that the principal must pay for access. Example F below expresses the condition that anyone can print the content "XrML book" upon payment of a fee of $3.10. Example G below expresses the condition that anyone can print the content "XrML book" upon payment of a fee of $3.10 for each print. Example H below expresses the condition that anyone can view the content "XrML book upon payment of a fee of $10.00 per hour of viewing time.

```
Example F:
<license>
        <inventory>
                <digitalWork licensePartID="XrMLBook"/>
        </inventory>
        <grant>
                        <forAll varName="anyone"/>
                        <principal varRef="anyone"/>
                        <print/>
                        <digitalWork licensePartId="XrMLBook"/>
                        <fee>
                                <paymentFlat>
                                <rate currency="USD">3.10</rate>
                                </paymmentFlat>
                                <to>
                                <aba>
                                <institution>123456789></institution>
                                <account>987654321</account>
                                </aba>
                                </to>
                </fee>
        </grant>
</license>

Example G:
<license>
        <inventory>
                <digitalWork licensePartID="XrMLBook"/>
        </inventory>
        <grant>
                <forAll varName="anyone"/>
                <principal varRef="anyone"/>
                <print/>
```

```
                <digitalWork licensePartIDRef="XrMLBook"/>
                <fee>
                        <paymentPerUse>
                        <rate currency="USD">3.10</rate>
                        </paymentPerUse>
                </fee>
        </grant>
</license>

Example H:
<license>
        <inventory>
                <digitalWork licensePartID="XrMLBook"/>
                </inventory>
        <grant>
                <forAll varName="anyone"/>
                <principal varRef="anyone"/>
                <view/>
                <digitalWork licensePartIDRef="XrMLBook"/>
                <fee>
                                <paymentMetered>
                                <rate currency="USD">10.00</rate>
                                <per>PT1H</per>
                                <phase>PT10M</phase>
                                </paymementMetered>
                                <to>
                                <aba>
                                <institution>123456789></institution>
                                <account>987654321</account>
                                </aba>
                        </to>
                </fee>
        </grant>
        </license>
```

[0030] Example I below expresses the condition that anyone can print the content but the exercise of the print right will be tracked by a tracking service before printing.

```
Example I:
<license>
        <inventory>
                <digitalWork licensePartID="XrMLBook"/>
                <keyHolder licensePartID="Edgar"/>
        </inventory>
```

```
<grant>
    <forAll varName="anyone"/>
    <principal varRef="anyone"/>
    <print/>
    <digitalWork licensePartIDRef="XrMLBook"/>
    <trackReport>
        <stateReference>
            <uddi>
                <serviceKey>
                <uuid>...</uuid>
                </serviceKey>
                </uddi>
                <serviceParam>
                </serviceParam>
        </stateReference>
    </trackReport>
</grant>
</license>
```

[0031] Conditions 80 may also be conditions on the system in which resource 100 is consumed. For example, condition 80 may require that the system has an authorized security mechanism or other specific hardware or software, or only a specific maximum number of users are logged on.

[0032] Conditions 80 can specify the repository or other device in which resource 100, such as content, resides. Conditions 80 can relate to the approval notice that the principal must obtain before using the protected resources 100. Conditions 80 can require notification before or after using resource 100. Conditions 80 can specify the previous rights related to the protected resource 100 or other resources. Conditions 80 can also be imposed on other conditions 80 such as how to verify a condition.

[0033] Of course, conditions 80 are not limited to the above examples, but can be any restriction, obligation or requirement that is associated with a right to protected resource 100 as pre-conditions, during-access conditions, post-conditions or other conditions. Also, even though the examples above are expressed using XrML™ conditions are not limited to or by XrML and can be expressed in any manner.

[0034] Fig. 4 schematically illustrates condition 80 in accordance with the preferred embodiment. Condition 80 includes resource designation 82 which can be expressed either implicitly or explicitly. For example, in Example A above, resource designation 82 is indicated by the attribute "licensePartID" of the "digitalWork" elements . Condition 80 also includes state variables 84, and method specification 86 indicating how values of state variables 84 can be obtained. Method specification 86 can include the location(s) where values of state variables 84 are stored (such as a remote state controller that manages a condition as described below), the communication protocol to communicate with the state controller where the condition is managed, and any parameters needed to obtain the value (such as service parameters, etc). Alternatively, the method can be hard-coded in the system and Method Specification 86 can be omitted.

[0035] As noted above, state variables 84 represent the status of condition 80. Each state variable 44 has a corresponding value at any moment in time for the principal, right, and resource. As note above, the collection of current values of state variables for a given right is referred to as the "state of rights" herein. Fig. 5 illustrates state of rights 50 of the preferred embodiment which includes condition 80, and current values 52 for state variables 84 of condition 80. Method specification 56 indicates the method used to obtain current values 52 of state variables 84, including potentially a source from which the value is obtained, the digital signature of a credential, the session ID of the request and any other appropriate information. Note that method specification 56 can be considered redundant with method specification 86 and can merely be a reiteration thereof. However, in some cases, method specification 56, used to actually obtain values 52, can be different than method specification 86, which is suggested for use in condition 80.

[0036] Using state of rights 50 to represent condition 80 for a right simplifies the process of verifying conditions 80 because all the information needed to verify condition 80 is readily accessible. State of rights 50 is constructed and then used whenever the corresponding condition 80 is

evaluated and verified. Each state of rights 50 can contain all information needed to verify values 52 of state variables 84.

[0037] An authenticated principal is a user that has been processed by a system which validates the authenticity of that user, for example when a user successfully logs-in with a user name and a password, the user becomes an authenticated principal or just "principal". Condition 80 for a given set of rights is defined as a set of required states variable values under which a principal is allowed to access protected resource 100. When an authenticated principal wishes to access protected resource 100, the system state changes from an "original state" to an "authorized state".

[0038] Once the system is in an authorized state the principal is then able to access the protected resource 100 for an authorized operation. In many cases, it is not the authenticated principal itself that actually accesses protected resource 100. For example, the access can be delegated to another authenticated principal such as a rendering application, service, or the like. While protected resource 100 is being accessed and consumed, the set of pre-conditions 80 for granting the initial access may no longer be applicable for authorizing continued access. Also, consuming protected resource 100 may transform the resource into a set of temporary, .i.e., derived, resources 100a from which the access conditions 80 imposed on the original resources are also not applicable. In order to protect resource 100 and its derived resources 100a while they are being accessed, the preferred embodiment uses a concept for authorization and protection called "during-access conditions" which is described in detail below.

[0039] In a conventional system, resources are in one of two states. As illustrated in Fig. 2, when resource 100 is inactive, the system is in original state 102 until pre-conditions are satisfied. At that time, resource 100 becomes active and the system enters the authorized or active state 104. To increase control over resources, the preferred embodiment defines two additional states in addition to the "original state" and "authorized state". As

illustrated in Fig. 3, during the use or access of protected resource 100, the system state will change through the following states: original state 102, authorized state 104, usage state 106 and end state 108. Conditions 80 can be defined, for each state, that must be met in order to move to the next state or continue within the same state. As note above with respect to Fig. 1, conditions 80 can be defined and prepared using preparation application 70 including any necessary user interface and editing capabilities. Conditions 80 that need to be satisfied to enter the Authorized State are called "pre-conditions". Conditions 80 that need to be satisfied during the use of resource 100 are called "during-access conditions" and conditions 80 that are required at the end of the use are called "post-conditions". Condition validator 44 can invoke the requisite conditions 80 for each state.

[0040] During-access conditions are transferred from the original resource 100 to itself and any of derived resources 100a while they are being accessed and consumed by an authenticated principal. For example, if resource 100 is a document which is displayed on a screen of client environment 30 during the authorized operation "view," then derived resources 100a may include the memory that contains the data from the document, the presentation format of the document, and the displayed windows respectively. Derived resources 100a will all be protected by the set of during-access conditions. In other words, the principal will only have access to derived resources 100a as long as the during-access conditions are satisfied. During-access conditions can be defined in the same manner as other conditions 80.

[0041] Another example is where an application, such as application 12, requests a service that is a protected resource 100. Once the request is authorized, the application that executes the service may be considered as a derived resource and is subjected to the set of during-access conditions while the service is being executed. During-access conditions define authorized state of rights 44a are applied to current state of rights 44b in the manner described n detail below. Once the requested operation is completed, either mandatory or voluntarily, all derived resources 100a protected by during-

access conditions are deleted (or disabled) and the system state is then transferred to the final state by the set of post-access conditions.

[0042] Conditions 80 after or during the use or access of a resource may or may not change. Those conditions with unchanged state are called "stateless conditions", while conditions that change after or during the use of the resource are called "stateful conditions". Pre-conditions 80 are usually stateless conditions 80 and are used to control the access to the protected document. During-access conditions and post-conditions are usually stateful conditions 80. They are used to control the lifetime of protected resource 100. (For example, protected resource 100 can no longer be accessed once a specific number of users logged into a network has been exceeded.) With these expanded types of conditions 80 associated with different stages of protected resource 100, the preferred embodiment provides a mechanism to authorize the use of protected resource 100 and to protect and track resource 100 while it is being used.

[0043] As illustrated in Fig. 3 and with reference to elements of Fig. 1, a system in accordance with the preferred embodiment progresses through three stages. Access authorization stage 302 is a stage in which state manager 40 authorizes an authenticated principal to access protected resource 100 for an authorized operation through verifying that pre-conditions are satisfied. Resource protection stage 304 is a stage in which state manager 40 protects resource 100 and its derived resources 100a while they are being used by verifying that the during-access conditions remained satisfied. Operation termination stage 306 is a stage in which state manager 40 terminates the use of protected resource 100 and derived resources 100a for a given operation when post-access conditions are satisfied or during-access conditions otherwise cease being satisfied.

[0044] In recursive situations, in which multiple accesses are given for the same resource 100, the post-condition can be the same as the pre-condition of the next cycle. In such a case, a non-static parameter can be used in order

to prevent infinite loop situations. For example a time-dependent condition or a condition modified or imposed by an external entity, such as human intervention.

[0045] Access authorization grants an authenticated principal the right(s) to access protected resource 100 for an authorized operation. Pre-conditions are evaluated, i.e., enforced, against the state of rights. If enforcement is successful, resource 100 and any derived resources 100a enter the authorized state in step 406 and during-access conditions begin to be enforced.

[0046] As noted above, resource protection protects both the initial protected resource 100 and its derived resources, 100a by enforcing the set of during-access conditions. The authorized state returned from the access authorization state contains the list of during-access conditions to be enforced during the authorized operation. In a mandatory system all derived resources 100a can be registered with resource repository 46 (see fig. 7 and description below) of resource manager 42 when derived resources 100a are generated and used. If any during-access condition becomes invalid resource manager 42 will disable access to protected resource 100 and derived resources 100a by application 12.

[0047] The termination of an authorized operation executes the set of post conditions (if any are present). Executing post-conditions can permanently change the state of rights and affects the next request for access to resource 100. For example, if a post condition is the removal of access to the resource 100 after an exercise limit has been reached, when the limit is reached the resource 100 is deleted or some other action is taken which disables or prevents access. Operation termination can include resource termination. Resource manager 42 can delete (disable) derived resources 100a when the operation is being terminated, whether or not the operation is forced to terminate, or the application is voluntarily terminating the operation. Deletion (or disabling) of derived resource 100a is important in the process of

protection of resource 100. Condition validator 44 commits the use of protected resource 100 and enforce the set of post-conditions. Condition validator 44 will invalidate (disable) protected resource 100 if resource 100 becomes invalid as a result of the change in the system state.

[0048] It can be seen that the state of rights may change during the use of resource 100 and therefore, there is a need to maintain, update and retrieve the state of rights. As noted above, condition validator 44 accesses current state of rights 44b to validate conditions 80 that are associated with rights. While resources 100 are being used or accessed, Condition validator 44 monitors the set of during-conditions and manages the current state of rights 44b. Condition validator 44 interacts with resource manager 42 to control derived resource 100a. When the current state of rights is no longer valid, i.e., no longer satisfies the during-access conditions the during-condition validator 44 requests that resource manager 42 delete (or disable) all the derived resources 100a or to notify application 12 that the use of derived resources 100a is no longer allowed. Using the state of rights to represent condition 80 simplifies the process of verifying conditions 80 because all the information needed to verify a specific condition 80 is readily available.

[0049] As illustrated in Figs. 1 and 6, state manager 40 also includes state of rights framework 20 which provides the API to condition validators 44 and provides a plug-in infrastructure to state controller 22 and state validator 24. Framework 20 handles the communication and coordination of tasks among plural state controllers 22 and validators 24.

[0050] Each state controller 22 is a component that manages the state for, i.e., tracks the value of a given state variable. The basic structure of state controller 22 consists of a software component that manipulates APIs defined by state of rights framework 20, a protocol to interact with the persistent storages or services to store, and update and query for the current value of the state variable and the history of the state variable. The location of the

persistent storage or service that manages the state variable is thus transparent to the state of rights framework 20.

[0051] Each state validator 24 is a component that verifies the value of a state variable. Each state validator 24 includes a software component that implements the set of interfaces defined by state validation and monitoring API 20c of rights framework 20. Like state controller 22 the state validator 24 may operate locally or remotely.

[0052] As illustrated in Fig. 6, state of rights framework 20 includes state manipulation API 20b which is a set of interfaces to initialize, query, update and transfer state of rights values. As noted above, the basic structure for condition 80 includes the state variables and a method needed to obtain the current value of the state variables.

[0053] Like state variables, the values can be represented by a grammar such as a data structure or the XrML™ rights language that describes the current value. The preferred embodiment uses XrML™ to define the state variables and extensions thereof to define the value of the state variables. However the representation of the state variables and its values are not limited to or by XrML™. The following example, shows how a state variable can be defined using XrML™.

Example 5:

```
<print/>
<sx:stateReference>
   <uddi>
      <serviceKey>
         <uuid>1F8903B0-FC03-4c5b-A445-
AAFCCEC011111</uuid>
      </serviceKey>
   </uddi>
</sx:stateReference>
<sx:exerciseLimit>9</sx:exerciseLimit
```

[0054] The above example describes the state variable of a print right that allows no more than 9 copies of the resource to be printed. The value

associated with a condition could be as simple as a number as in an example of an exercise limit, or could be a boolean value "yes" or "no" as in an example of requiring an approval. The value can also be stored and managed by a service or component as shown in example K below.

```
Example K:
<play/>
        <validityIntervalFloating>
            <sx:stateReference>
                <uddi>
                    <serviceKey>
                        <uuid>1F8903B0-FC03-4c5b-A445-
AAFCCEC011111</uuid>
                    </serviceKey>
                </uddi>
            </sx:stateReference>
        </validityIntervalFloating>
```

[0055] Example K illustrates a state associated with a floating interval of a "play" right. In this example, the state variable does not contain an actual value associated with the condition. Rather, the remote service (stateReference), acting as state controller 22 is the one that manages the value and it is "opaque" in the expression for the state variable.

[0056] State manipulation API 20b includes a query interface to query the values of a state variable. A query for a value state variable requires an input that contains the state variable to be queried and the value-type of the response. The value of the response for the state query can be the state value, the state history or both.

[0057] The following example L the value of a state variable "exerciseLimit" associated with a "print" right.

```
Example L:
<stateQuery>
            <! State or the exerciseLimit condition for a print rights >
            <print/>
             <sx:stateReference>
                <uddi>
```

```
                        <serviceKey>
                            <uuid>1F8903B0-FC03-4c5b-A445-
AAFCCEC011111</uuid>
                        </serviceKey>
                    </uddi>
                </sx:stateReference>
                <sx:exerciseLimit/>
                    <! Request for a specific response type>
            <response type="value">
        </stateQuery>
```

[0058] In order to process the query, the state of rights framework 20 will determine what state controller 22 is responsible for this request and then locate, authenticate and load that specific state controller 22 and pass the request to the state controller 22 for processing. State controller 22 can be local or remote. Once the request is processed the response is then returned to the requester via the state of rights framework 20. The following example M describes the response

Example M:

```
<stateResponse>
            <! Response to the following query request>
    <stateQuery>
            <print/>
              <sx:stateReference>
                  <uddi>
                      <serviceKey>
                          <uuid>1F8903B0-FC03-4c5b-A445-
AAFCCEC011111</uuid>
                      </serviceKey>
                  </uddi>
              </sx:stateReference>
               <sx:exerciseLimit>9</sx:exerciseLimt>
        </stateQuery>
          <! State value returned from state query>
        <stateValue>8</stateValue>
          <! Digital signature of the credential >
        <dsig:signature>
            ...
        </dsig:signature>
</stateResponse>
```

[0059] The query response contains the original query, current value of the queried state variable if available, current status of the state variable or the state history, an identification, a session id and a digital signature of the response. The digital signature can be used guarantee the integrity of the response.

[0060] The state can only be updated based on the prior response of the state query provided that the "sessionID" in the state query's response is valid. The sessionID is used in the preferred embodiment to identify a request, but any other identification can be used to match the query and the update. Thus, state manage 40 must query the current value of the state variable to obtain the state variable value and the valid session ID in order to update the target state variable. Updating the value of state variable will change the current value of the state variable to the new value.

[0061] There are many constraints that can be imposed on updating the value of the state variable. For example, the rights associated with the state variable must still be valid prior to the update request. Another constraint can be that the new value of the state variable after updating must be a valid value allowed for the state variable. For example if the maximum print copy (state variable) is 4 and the current print copy is 3 (current value of state variable) then a request for 2 more copies (update value) will be rejected because the value of the state variable after updating is not an allowed value. Additionally, The principal or the application making the request for the update must be authorized to do so. State of rights framework 20 will identify, authenticate and load state controller 22 to process an updating request. Example M below is an example of a request to update a state value.

Example M:

```
<stateUpdate>
    <! Update on the previous query value. If the query does not
match with >
    <! The current value of the state variable the update request will
be fail >
     <stateQuery>
```

```
                    <sx:stateReference>
                        <uddi>
                            <serviceKey>
                                <uuid>1F8903B0-FC03-4c5b-A445-
AAFCCEC011111</uuid>
                            </serviceKey>
                        </uddi>
                    </sx:stateReference>
                    <sx:notMoreThan>9</sx:notMoreThan>
                  <! Session Id is implementation specific to match the
updated request >
                   < and the previous query. Any other identification scheme
can be used instead >
                  <sessionID>...</sessionID>
                   <stateController>
                       <id>...</id>
                   </stateController>
                   <stateValue>8</stateValue>
                   <dsig:signature>
                      ...
                   </dsig:signature>
              </stateQuery>
             <updateValue>1</updateValue>
        </stateUpdate>
```

[0062] State value transfer is similar to the state value update in terms of state management. The main different between the two is how the state history for the state variable is maintained. Once the state is transferred, the current value of the state variable and its history are updated according to the transfer. If the rights from which its states are transferred expire as the result of the state transfer then the transfer is called a complete transfer, otherwise it is called a partial transfer. A state value transfer can have the same constraint as the state value update. The elements involved in the transfer of the state value can be authorized before the transfer occurs.

[0063] State manager 40 also manages a set of state validators 24 and provides the set of interfaces to the application which verifies each state variable individually or as a set representing the state of rights. Recall that state of rights is the collection of current values of state variables associated with a given right. State of rights framework 20 will select and load the

appropriate state validator 24 for each of the state variables. Once the state validator 24 is selected and loaded, state of rights framework 20 then passes the request to the target state validator 24 and waits for the result as a response message. State manager 40 may concurrently execute all state validators 24 at once or sequentially execute them depending on the dependency between state variables and the configuration of the system.

[0064] State validator 24 verifies the state variable value given a state query response. Upon receiving the request for validation for a state variable, state of rights framework 20 examines the state query response and then selects, authenticates and loads the appropriate state validator 24 and passes the state query response to the state validator 24. Based on a configurable policy and information stored in the state query state validator 24 may accept or challenge the information stored in both the state query and response. Once the information stored in the state query and response are verified, the validation process can be as simple as comparing the current value of the state variable with the set of possible (or allowed) values of the state variable. Each state validator 24 can include a software component that implements the set of interfaces defined by state of rights framework 20. State validator 24 may operate locally or remotely.

[0065] The preferred embodiment can utilize various devices, such as a personal computers, servers, workstations, PDA's, thin clients and the like. For example, the client environment can be a handheld device such as a mobile phone or a PDA. Various channels for communication can be used. Further, the various functions can be integrated in one device. The disclosed functional devices and modules are segregated by function for clarity. However, the various functions can be combined or segregated as hardware and/or software modules and devices in any manner. The various functions modules and devices have utility separately or in combination.

[0066] The various records, messages, elements and portions thereof can be stored on the same device or on different devices. Various links,

references, specifications, and the like can be used to associate the elements. Access to any type of resource can be controlled. Any mechanism for tracking values of state variables can be used.

[0067] The invention has been described through a preferred embodiment and examples. However, various modifications can be made without departing from the scope of the invention as define by the appended claims and legal equivalents.

1. A system for managing the state of a protected resource in a system for granting access to a protected resource in accordance with usage rights, said usage rights including state variables indicating a status of an associated protected resource, said system comprising:

a protected resource associated with a usage right defined at least in part by a state variable;

a resource control device coupled to said resource to control use of said resource by enforcing the usage right;

a state controller operative to track the value of a state variable; and

an interface framework operative to receive a message related to said state variable from said resource management device, load said state controller, and instruct said state controller to manipulate the value of the state variable in accordance with said message.

2. A system as recited in claim 1, wherein the message is a value query and wherein said interface framework is operative to instruct said state controller to retrieve a current value of the state variable and return the value as a query response.

3. A system as recited in claim 1, wherein the query response includes the current value, the query, and an ID mechanism.

4. A system as recited in claim 3, wherein said resource control is operative to update a stored value of the state variable with the current value.

5. A system as recited in claim 1, wherein there are plural state controllers corresponding to plural state variables.

6. A system as recited in claim 1, further comprising at least one derived resource for exercising the usage right of the protected resource, said state variables being transferred to said protected resource.

7. A system as recited in claim 2, further comprising a state validator operative to apply logic to the query response to validate the query response.

8. A system as recited in claim 4, wherein said resource control device comprises a condition validator operative to enforce conditions of the usage right against the stored value.

9. A method for managing the state of a protected resource in a system for granting access to a protected resource in accordance with usage rights, said usage rights including a state variable indicating a status of an associated protected resource, said method comprising:

transmitting a message related to the state variable from a resource control device to an interface framework, said resource control device being coupled to said resource to control use of said resource by enforcing the usage right;

loading into said framework a state controller operative to track the value of the state variable; and

instructing said state controller to manipulate the value of the state variable in accordance with said message.

10. A method as recited in claim 9, wherein the message is a value query and wherein said instructing step comprises instructing said state controller to retrieve a current value of the state variable and return the value as a query response.

11. A method as recited in claim 9, wherein the query response includes the current value, the query, and an ID mechanism.

12. A method as recited in claim 11, further comprising updating a stored value of the state variable in said resource control device with the current value.

13. A method as recited in claim 9, further comprising deriving at least one derived resource for exercising the usage right of the protected resource, and transferring said state variables to said protected resource.

14. A method as recited in claim 9, further comprising applying logic to the query response to validate the query response.

15. A method as recited in claim 12, further comprising enforcing conditions of the usage right against the stored value.

Fig. 1

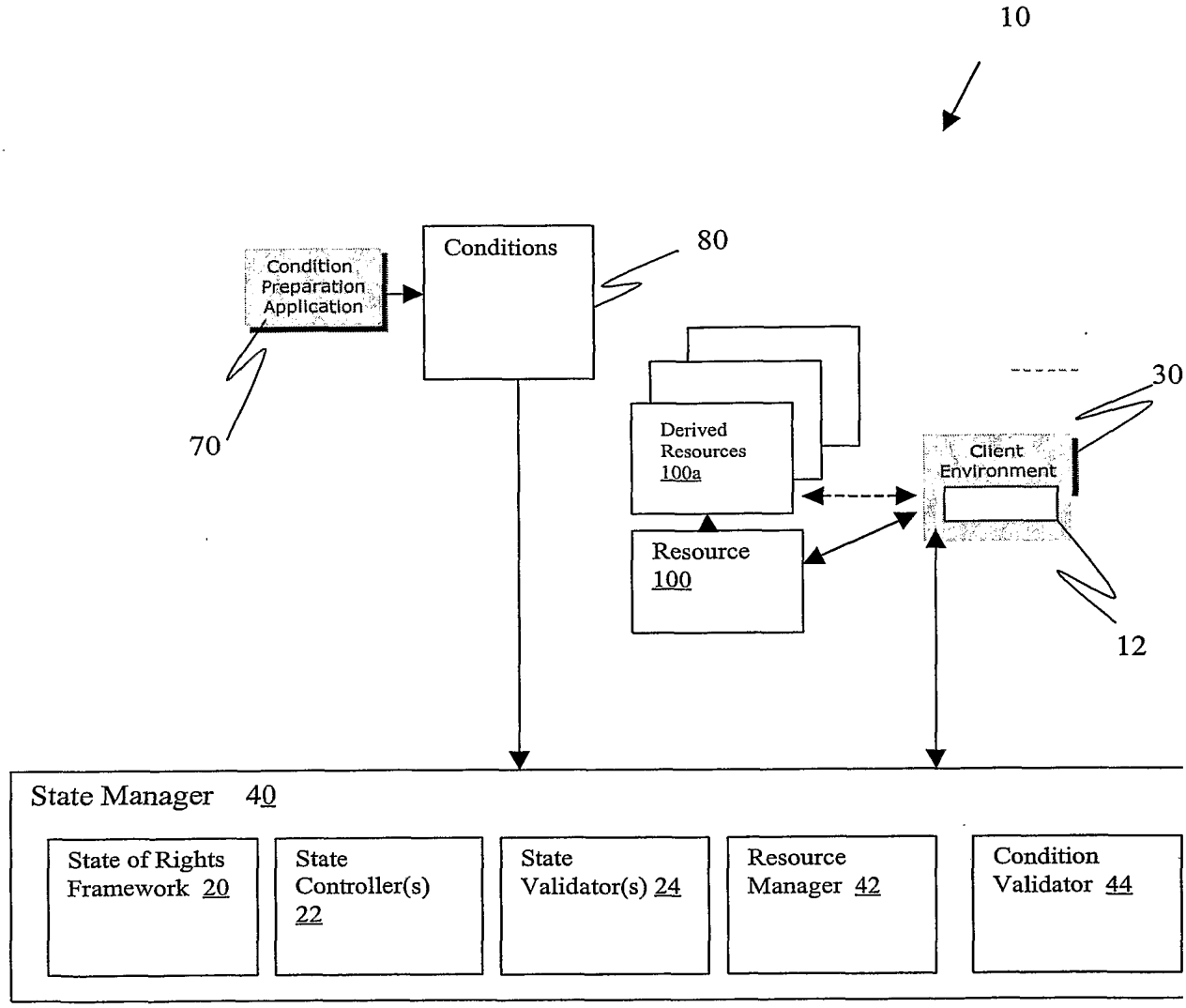

10
Condition Preparation Application
Conditions
80
70
Derived Resources 100a
Client Environment
30
Resource 100
12
State Manager 40
State of Rights Framework 20
State Controller(s) 22
State Validator(s) 24
Resource Manager 42
Condition Validator 44

Fig. 2

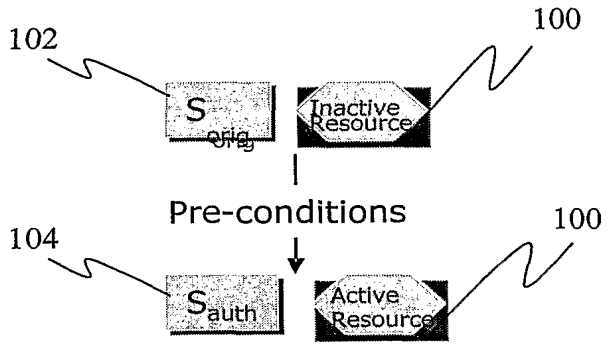

102
100
S_{orig}
Inactive Resource
Pre-conditions
104
100
S_{auth}
Active Resource

Fig. 3

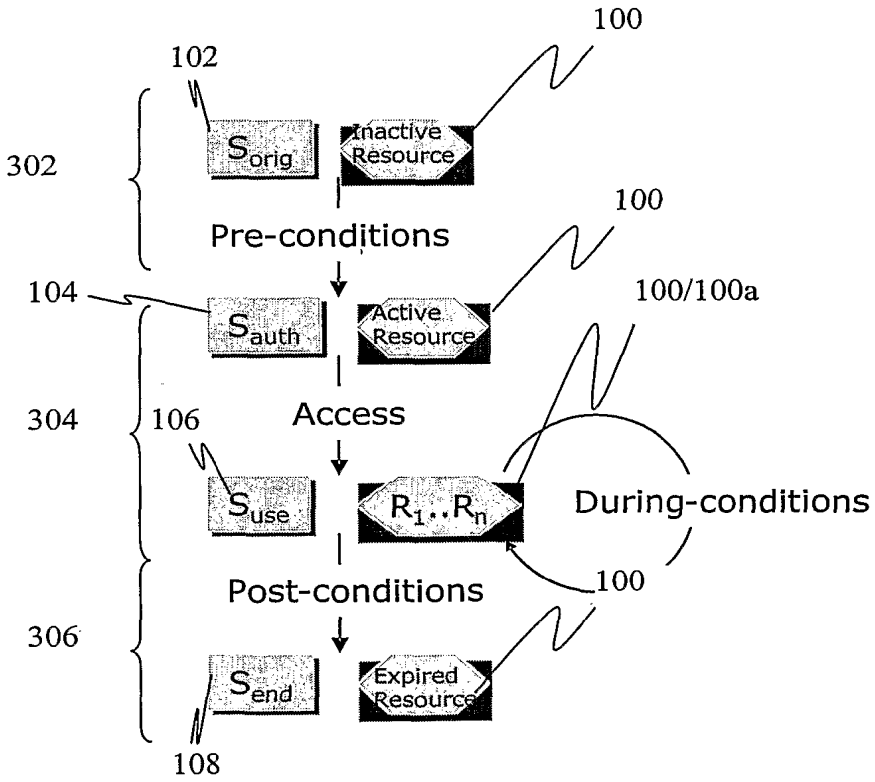

100
102
302
S_{orig}
Inactive Resource
Pre-conditions
100
104
S_{auth}
Active Resource
100/100a
304
106
Access
S_{use}
$R_1..R_n$
During-conditions
Post-conditions
100
306
S_{end}
Expired Resource
108

Fig. 4

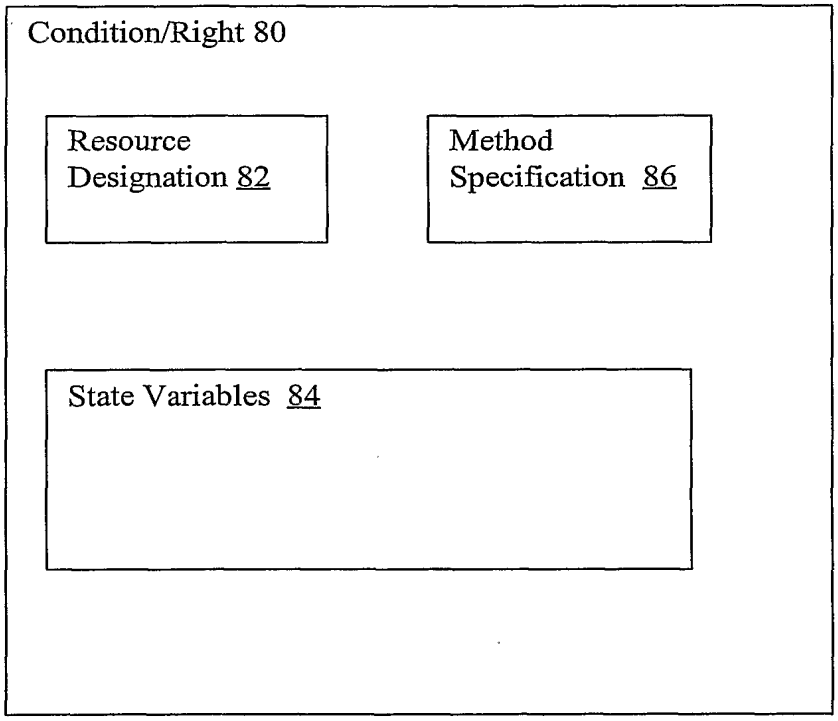

Condition/Right 80
Resource Designation 82
Method Specification 86
State Variables 84

Fig. 5

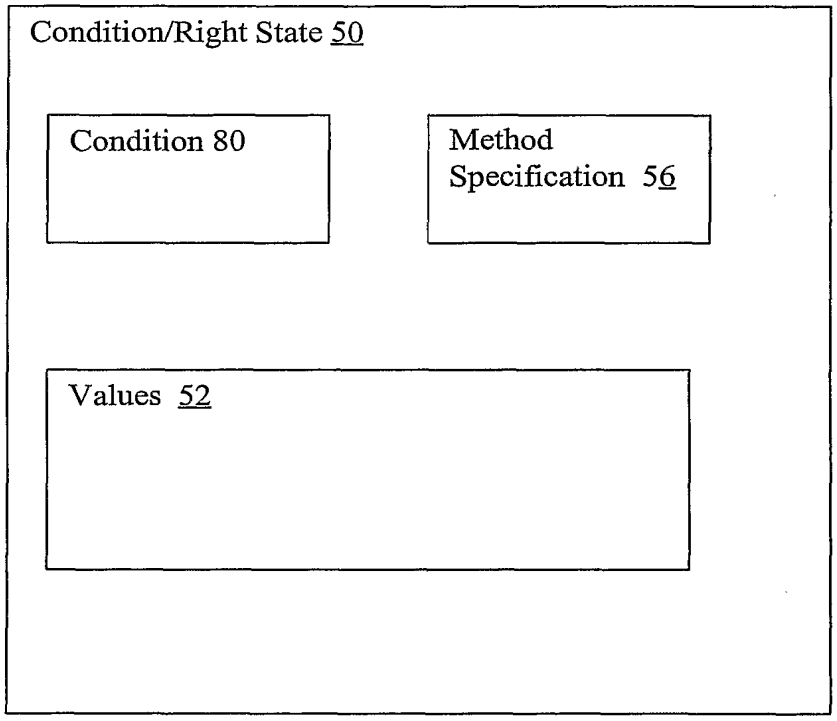

Condition/Right State 50
Condition 80
Method Specification 56
Values 52

Fig. 6

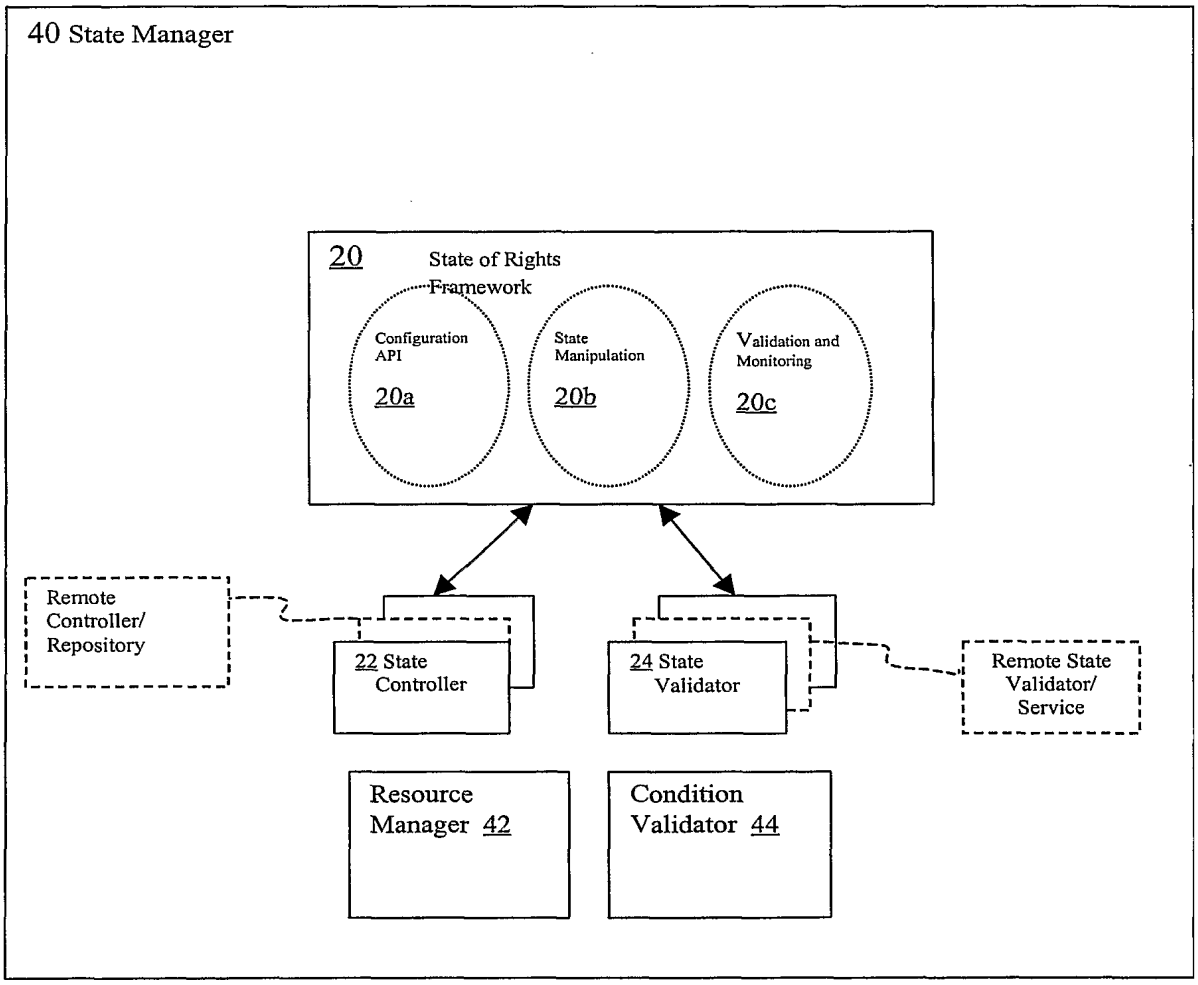
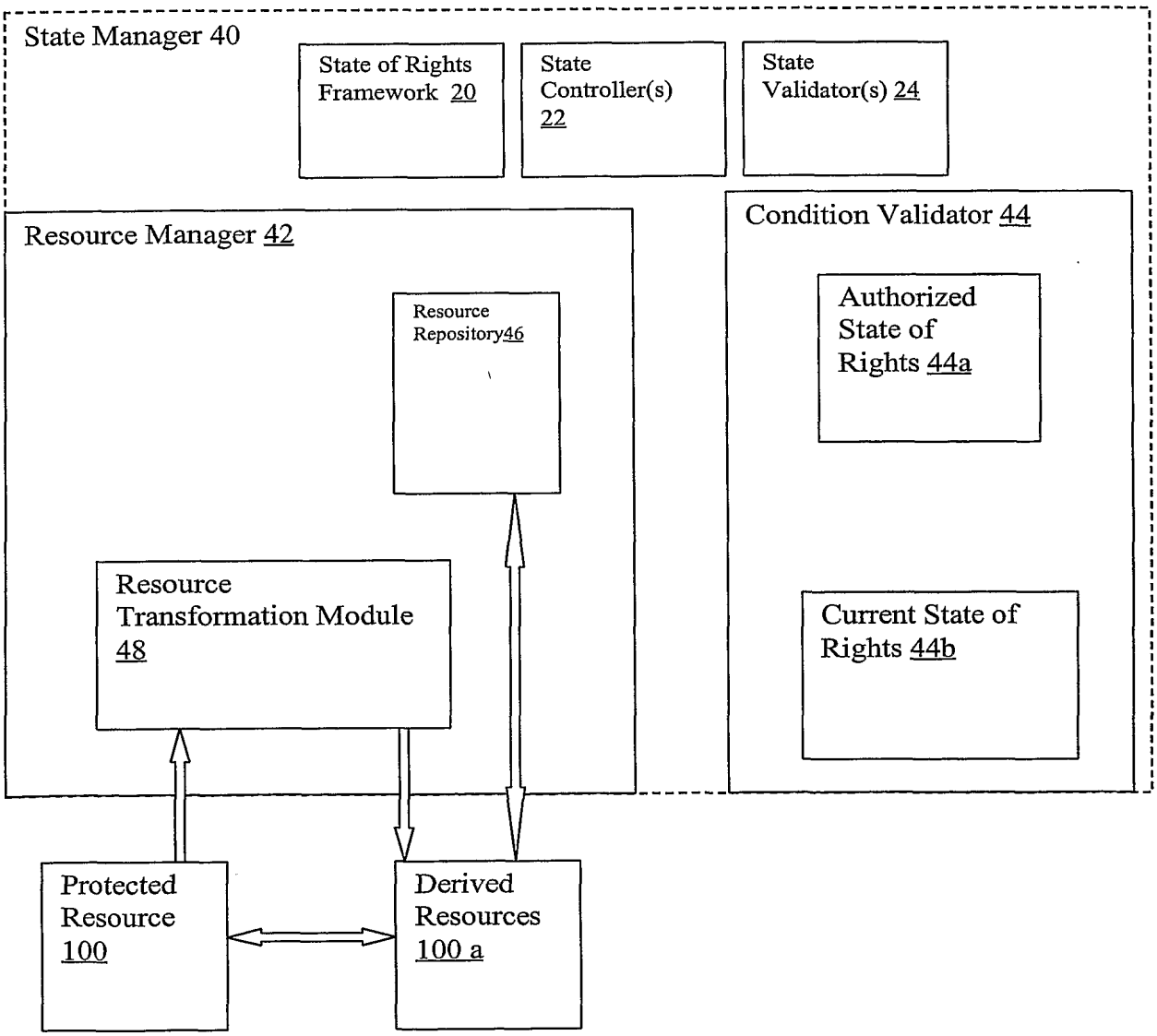

40 State Manager
20
State of Rights Framework
Configuration API
20a
State Manipulation
20b
Validation and Monitoring
20c
Remote Controller/ Repository
22 State Controller
24 State Validator
Remote State Validator/ Service
Resource Manager 42
Condition Validator 44

Fig. 7


State Manager 40
State of Rights Framework 20
State Controller(s) 22
State Validator(s) 24
Resource Manager 42
Resource Repository46
Resource Transformation Module 48
Condition Validator 44
Authorized State of Rights 44a
Current State of Rights 44b
Protected Resource 100
Derived Resources 100 a